# Passeio Aleatório

May 6, 2025

## Contents

# 1 Passeio Aleatório

Na teoria das probabilidades, um **passeio aleatório** é um modelo que representa uma sequência de passos aleatórios ao longo do tempo. Ele é amplamente utilizado para descrever fenômenos estocásticos em diversas áreas, como física, biologia, finanças e ciência de dados.

## 1.1 Passeio em $\mathbb{Z}$

Considere uma variável aleatória $y_i$, **i.i.d.** (*independente e identicamente distribuída*), definida por:

$$y_i = \begin{cases} 1, \text{ com } p(1) = \frac{1}{2} \\ -1, \text{ com } p(-1) = \frac{1}{2} \end{cases}$$

Definimos então a posição acumulada no tempo $t$ como:

$$X_T = \sum_{i=1}^{T} y_i, \ \ com \ X_o = 0$$

- $X_0$ é considerado o estado inicial

A sequência $X_0, X_1, X_2, ..., X_t$ constitui o que chamamos de **passeio aleatório**.

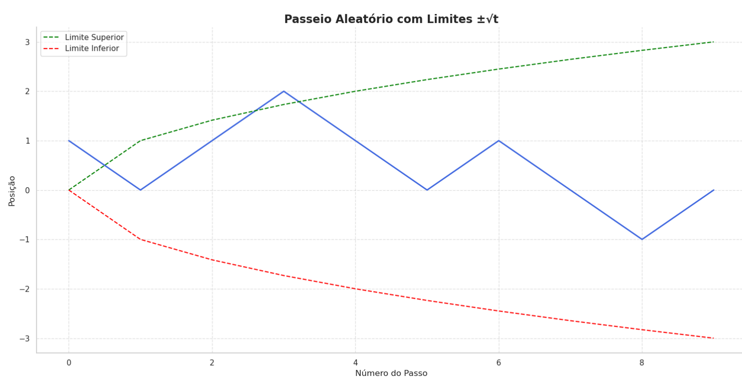


Figure 1: randomwalk.png

### 1.1.1 Simulação em Python

A seguir, uma simulação simples de um passeio aleatório unidimensional:

```python
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

def randomWalk(n):
    """
    Simula um passeio aleatório em uma dimensão.

    Parâmetros:
    n (int): Número de passos do passeio aleatório.

    Retorna:
    list: Lista com as posições ao longo do passeio.
```

```
    """
    steps = np.random.choice([-1, 1], size=n)
    positions = np.cumsum(steps)
    return positions.tolist()

def plotRandomWalk(positions):
    """
    Plota o passeio aleatório com limites dinâmicos +/- sqrt(t).
    """
    sns.set(style="whitegrid")
    plt.figure(figsize=(12, 6))

    steps = range(len(positions))
    upper_bound = [np.sqrt(t) for t in steps]
    lower_bound = [-np.sqrt(t) for t in steps]

    sns.lineplot(x=steps, y=positions, color="royalblue", linewidth=2)

    plt.plot(
        steps,
        upper_bound,
        color="green",
        linestyle="--",
        linewidth=1.5,
        label="Limite Superior",
    )
    plt.plot(
        steps,
        lower_bound,
        color="red",
        linestyle="--",
        linewidth=1.5,
        label="Limite Inferior",
    )

    plt.title("Passeio Aleatório com Limites ±sqrt(t)", fontsize=16, weight="bold")
    plt.xlabel("Número do Passo", fontsize=12)
    plt.ylabel("Posição", fontsize=12)
    plt.grid(True, linestyle="--", alpha=0.6)
    sns.despine()
    plt.legend()
    plt.tight_layout()
    plt.show()

walk = randomWalk(10)
# print("Posições do passeio aleatório:", walk)
plotRandomWalk(walk)
```

Sabendo disso, podemos calcular a média e a variância de $y_i$ e de $X_T$:

$$\mathbb{E}[y_i] = 1 \cdot \frac{1}{2} + (-1) \cdot \frac{1}{2} = 0, \quad \mathrm{Var}[y_i] = 1^2 \cdot \frac{1}{2} + (-1)^2 \cdot \frac{1}{2} = 1, \quad \sigma = \sqrt{1} = 1$$

$$\mathbb{E}[X_T] = \mathbb{E}\left[\sum_{i=1}^{T} y_i\right] = \sum i = 1^T \mathbb{E}[y_i] = 0, \quad \mathrm{Var}[X_T] = \mathrm{Var}\left[\sum_{i=1}^{T} y_i\right] = \sum i = 1^T \mathrm{Var}[y_i] = T, \quad \sigma = \sqrt{T}$$

Podemos observar que a esperança do passeio aleatório simples é sempre zero, pois trata-se de um processo simétrico — sem tendência para valores positivos ou negativos. Além disso, a variância de $X_T$ cresce linearmente com o tempo, já que cada passo contribui de forma independente e com variância constante. Por isso, o desvio padrão é dado por $\sqrt{T}$, justificando o uso dos limites $\pm\sqrt{T}$ na simulação visual apresentada anteriormente.

## 1.2 Passeio em $\mathbb{Z}^2$

Passeios aleatórios bidimensionais são amplamente utilizados em modelos epidemiológicos, pois permitem simular o deslocamento de um indivíduo contaminado em uma região espacial definida por coordenadas $(x, y)$.

Neste contexto, consideremos o seguinte exemplo: uma pessoa infectada, representada por $X_i$, inicia no ponto $(0, 0)$ e pode se mover para cima, para baixo, para a esquerda ou para a direita, cada direção com probabilidade igual a $\frac{1}{4}$.

$$p(x_i = (1,0)) = \frac{1}{4}, \quad p(x_i = (-1,0)) = \frac{1}{4}, \quad p(x_i = (0,1)) = \frac{1}{4}, \quad p(x_i = (0,-1)) = \frac{1}{4}$$

Para fins de simplificação, vamos assumir que a probabilidade de contaminação em qualquer local visitado é de 100%. Assim, podemos simular esse passeio aleatório e visualizar graficamente as áreas potencialmente contaminadas ao longo do tempo.

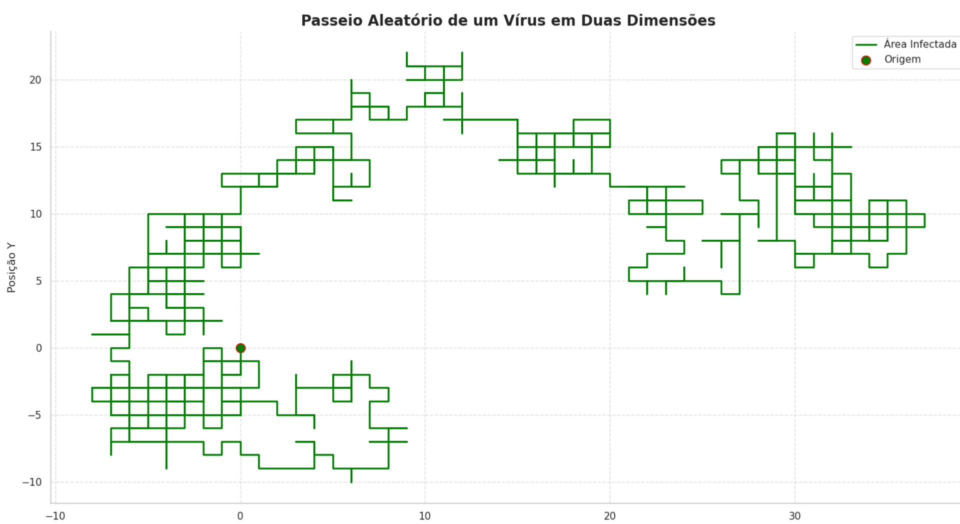


Figure 2: Passeio Aleatório Vírus

### 1.2.1 Simulação em Python

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

def virusWalk(n):
```

```
    """
    Simula um passeio aleatório em duas dimensões.
    O passeio é representado por um vetor de posições (x, y) ao longo do tempo.
    Cada passo é dado em uma direção aleatória (cima, baixo, esquerda, direita).
    Cada passo tem um tamanho fixo de 1 unidade.
    O passeio é iniciado na posição (0, 0).

    Parâmetros:
    n (int): Número de passos do passeio aleatório.

    Retorna:
    list: Lista com as posições ao longo do passeio.
    """
    # Inicializa a posição inicial
    position = np.array([0, 0])
    positions = [position.copy()]

    # Define as direções possíveis (cima, baixo, esquerda, direita)
    directions = np.array([[1, 0], [-1, 0], [0, 1], [0, -1]])

    # Realiza o passeio aleatório
    for _ in range(n):
        step = np.random.choice(range(4))
        position += directions[step]
        positions.append(position.copy())

    return positions

def plotDrunkWalk(positions):
    """
    Plota o passeio aleatório em duas dimensões.
    """
    sns.set(style="whitegrid")
    plt.figure(figsize=(12, 12))

    # Converte a lista de posições para um array numpy
    positions = np.array(positions)

    # Plota o passeio
    plt.plot(positions[:, 0], positions[:, 1], color="green", linewidth=2, label="Área Infectada")

    # Plota a origem
    plt.scatter(0, 0, facecolor="green",edgecolor='red', s=100, label="Origem")

    plt.title("Passeio Aleatório de um Vírus em Duas Dimensões", fontsize=16, weight="bold")
    plt.xlabel("Posição X", fontsize=12)
    plt.ylabel("Posição Y", fontsize=12)
    plt.grid(True, linestyle="--", alpha=0.6)
    sns.despine()
    plt.legend()
    plt.tight_layout()
    plt.show()
```

```
walk = virusWalk(1000)
plotDrunkWalk(walk)
```

## 1.3 Propriedades Fundamentais do Passeio Aleatório

1. **Média (Esperança)**
   - Para cada passo: $\mathbb{E}[y_i] = 0$
   - Para a soma dos passos: $\mathbb{E}[X_T] = \mathbb{E}\left[\sum_{i=1}^{T} y_i\right] = 0$
   - O passeio não tem tendência (é **não tendencioso**).
2. **Variância**
   - Para cada passo: $\mathrm{Var}[y_i] = 1$
   - Para a soma: $\mathrm{Var}[X_T] = \sum_{i=1}^{T} \mathrm{Var}[y_i] = T$
3. **Desvio Padrão**
   - $\sigma_{X_T} = \sqrt{T}$
   - A dispersão aumenta com o tempo.
4. **Simetria**
   - A distribuição de $X_T$ é simétrica em torno de 0.
   - Isto significa que $P(X_T = a) = P(X_T = -a)$
5. **Distribuição Limite**
   - Pelo **Teorema Central do Limite**, quando $T$ é grande: $\frac{X_T}{\sqrt{T}} \to \mathcal{N}(0, 1)$
   - Ou seja, o passeio se aproxima de uma normal padrão.
6. **Independência dos Passos**
   - Cada passo $y_i$ é independente dos outros.
   - Isso permite somar variâncias diretamente.
7. **Markoviano**
   - O processo é um **Processo de Markov**: $P(X_{t+1}|X_t, X_{t-1}, \ldots, X_0) = P(X_{t+1}|X_t)$
   - O futuro depende apenas do presente.
8. **Limite de Dispersão Típica**
   - Cerca de 95% das trajetórias ficam dentro de: $[-2\sqrt{T}, +2\sqrt{T}]$